\

All IBM PC and compatible computers are typically equipped with two serial ports and one parallel port. Although these two types of ports are used for communicating with external devices, they work in different ways.

A parallel port sends and receives data eight bits at a time over 8 separate wires. This allows data to be transferred very quickly; however, the cable required is more bulky because of the number of individual wires it must contain. Parallel ports are typically used to connect a PC to a printer and are rarely used for much else. A serial port sends and receives data one bit at a time over one wire. While it takes eight times as long to transfer each byte of data this way, only a few wires are required. In fact, two-way (full duplex) communications is possible with only three separate wires - one to send, one to receive, and a common signal ground wire.

[Bi-directional Communications](http://www.taltech.com/resources/intro-sc.html#bidir)
[Baud Versus Bits Per Second](http://www.taltech.com/resources/intro-sc.html#baud)
[Communicating By Bits](http://www.taltech.com/resources/intro-sc.html#commbits)
[Cables, Null Modems, And Gender Changers](http://www.taltech.com/resources/intro-sc.html#cablenulls)
[The Parity Bit](http://www.taltech.com/resources/intro-sc.html#parity)
[Cable Lengths](http://www.taltech.com/resources/intro-sc.html#length)
[RS-232C](http://www.taltech.com/resources/intro-sc.html#232c)
[Gender Changers](http://www.taltech.com/resources/intro-sc.html#gender)
[DCE And DTE Devices](http://www.taltech.com/resources/intro-sc.html#dce)
[Null Modem Cables and Adapters](http://www.taltech.com/resources/intro-sc.html#null)
[9 Pin To 25 Pin Adapters](http://www.taltech.com/resources/intro-sc.html#9to5)
[Synchronous And Asynchronous Communications](http://www.taltech.com/resources/intro-sc.html#Synch)
[Help! - My new computer does not have any COM Ports - What do I do now?](http://www.taltech.com/news/usbnocompt.htm)

**See Also:**

[Sartorius Cable Pinouts](http://www.taltech.com/support/dde_sw/sartopin.htm)
[Pinouts for other common types of cable](http://www.taltech.com/support/pins2.html)

--

|  |
| --- |
| **Bi-Directional Communications** |

The serial port on your PC is a full-duplex device meaning that it can send and receive data at the same time. In order to be able to do this, it uses separate lines for transmitting and receiving data. Some types of serial devices support only one-way communications and therefore use only two wires in the cable - the transmit line and the signal ground.
[Back to Top](http://www.taltech.com/resources/intro-sc.html#top)

--

|  |
| --- |
| **Communicating by Bits** |

Once the start bit has been sent, the transmitter sends the actual data bits. There may either be 5, 6, 7, or 8 data bits, depending on the number you have selected. Both receiver and the transmitter must agree on the number of data bits, as well as the baud rate. Almost all devices transmit data using either 7 or 8 databits.

Notice that when only 7 data bits are employed, you cannot send ASCII values greater than 127. Likewise, using 5 bits limits the highest possible value to 31. After the data has been transmitted, a stop bit is sent. A stop bit has a value of 1 - or a mark state - and it can be detected correctly even if the previous data bit also had a value of 1. This is accomplished by the stop bit's duration. Stop bits can be 1, 1.5, or 2 bit periods in length.
[Back to Top](http://www.taltech.com/resources/intro-sc.html#top)

--

|  |
| --- |
| **The Parity Bit** |

Besides the synchronization provided by the use of start and stop bits, an additional bit called a parity bit may optionally be transmitted along with the data. A parity bit affords a small amount of error checking, to help detect data corruption that might occur during transmission. You can choose either even parity, odd parity, mark parity, space parity or none at all. When even or odd parity is being used, the number of marks (logical 1 bits) in each data byte are counted, and a single bit is transmitted following the data bits to indicate whether the number of 1 bits just sent is even or odd.

For example, when even parity is chosen, the parity bit is transmitted with a value of 0 if the number of preceding marks is an even number. For the binary value of 0110 0011 the parity bit would be 0. If even parity were in effect and the binary number 1101 0110 were sent, then the parity bit would be 1. Odd parity is just the opposite, and the parity bit is 0 when the number of mark bits in the preceding word is an odd number. Parity error checking is very rudimentary. While it will tell you if there is a single bit error in the character, it doesn't show which bit was received in error. Also, if an even number of bits are in error then the parity bit would not reflect any error at all.

Mark parity means that the parity bit is always set to the mark signal condition and likewise space parity always sends the parity bit in the space signal condition. Since these two parity options serve no useful purpose whatsoever, they are almost never used.
[Back to Top](http://www.taltech.com/resources/intro-sc.html#top)

--

|  |
| --- |
| **RS-232C** |

RS-232 stands for Recommend Standard number 232 and C is the latest revision of the standard. The serial ports on most computers use a subset of the RS-232C standard. The full RS-232C standard specifies a 25-pin "D" connector of which 22 pins are used. Most of these pins are not needed for normal PC communications, and indeed, most new PCs are equipped with male D type connectors having only 9 pins.
[Back to Top](http://www.taltech.com/resources/intro-sc.html#top)

--

|  |
| --- |
| **DCE and DTE Devices** |

Two terms you should be familiar with are DTE and DCE. DTE stands for Data Terminal Equipment, and DCE stands for Data Communications Equipment. These terms are used to indicate the pin-out for the connectors on a device and the direction of the signals on the pins. Your computer is a DTE device, while most other devices are usually DCE devices.

If you have trouble keeping the two straight then replace the term "DTE device" with "your PC" and the term "DCE device" with "remote device" in the following discussion.

The RS-232 standard states that DTE devices use a 25-pin male connector, and DCE devices use a 25-pin female connector. You can therefore connect a DTE device to a DCE using a straight pin-for-pin connection. However, to connect two like devices, you must instead use a null modem cable. Null modem cables cross the transmit and receive lines in the cable, and are discussed later in this chapter. The listing below shows the connections and signal directions for both 25 and 9-pin connectors.

|  |
| --- |
| **25 Pin Connector on a DTE device (PC connection)** |
| Male RS232 DB25 | 25 pin to 9 pin connector |
| **Pin Number** | **Direction of signal:** |
| 1 | Protective Ground  |
| 2 | Transmitted Data (TD) Outgoing Data (from a DTE to a DCE) |
| 3 | Received Data (RD) Incoming Data (from a DCE to a DTE) |
| 4 | Request To Send (RTS) Outgoing flow control signal controlled by DTE |
| 5 | Clear To Send (CTS) Incoming flow control signal controlled by DCE |
| 6 | Data Set Ready (DSR) Incoming handshaking signal controlled by DCE |
| 7 | Signal Ground Common reference voltage |
| 8 | Carrier Detect (CD) Incoming signal from a modem |
| 20 | Data Terminal Ready (DTR) Outgoing handshaking signal controlled by DTE |
| 22 | Ring Indicator (RI) Incoming signal from a modem  |

|  |
| --- |
| **9 Pin Connector on a DTE device (PC connection)** |
| Male RS232 DB9  | 9 pin D connector |
| **Pin Number** | **Direction of signal:** |
| 1 | Carrier Detect (CD) (from DCE) Incoming signal from a modem |
| 2 | Received Data (RD) Incoming Data from a DCE |
| 3 | Transmitted Data (TD) Outgoing Data to a DCE |
| 4 | Data Terminal Ready (DTR) Outgoing handshaking signal  |
| 5 | Signal Ground Common reference voltage |
| 6 | Data Set Ready (DSR) Incoming handshaking signal |
| 7 | Request To Send (RTS) Outgoing flow control signal |
| 8 | Clear To Send (CTS) Incoming flow control signal |
| 9 | Ring Indicator (RI) (from DCE) Incoming signal from a modem  |

The TD (transmit data) wire is the one through which data from a DTE device is transmitted to a DCE device. This name can be deceiving, because this wire is used by a DCE device to receive its data. The TD line is kept in a mark condition by the DTE device when it is idle. The RD (receive data) wire is the one on which data is received by a DTE device, and the DCE device keeps this line in a mark condition when idle.

**RTS** stands for **Request To Send**. This line and the CTS line are used when "hardware flow control" is enabled in both the DTE and DCE devices. The DTE device puts this line in a mark condition to tell the remote device that it is ready and able to receive data. If the DTE device is not able to receive data (typically because its receive buffer is almost full), it will put this line in the space condition as a signal to the DCE to stop sending data. When the DTE device is ready to receive more data (i.e. after data has been removed from its receive buffer), it will place this line back in the mark condition. The complement of the RTS wire is CTS, which stands for Clear To Send. The DCE device puts this line in a mark condition to tell the DTE device that it is ready to receive the data. Likewise, if the DCE device is unable to receive data, it will place this line in the space condition. Together, these two lines make up what is called RTS/CTS or "hardware" flow control. The Software Wedge supports this type of flow control, as well as Xon/XOff or "software" flow control. Software flow control uses special control characters transmitted from one device to another to tell the other device to stop or start sending data. With software flow control the RTS and CTS lines are not used.

**DTR** stands for **Data Terminal Ready**. Its intended function is very similar to the RTS line. DSR (Data Set Ready) is the companion to DTR in the same way that CTS is to RTS. Some serial devices use DTR and DSR as signals to simply confirm that a device is connected and is turned on. The Software Wedge sets DTR to the mark state when the serial port is opened and leaves it in that state until the port is closed. The DTR and DSR lines were originally designed to provide an alternate method of hardware handshaking. It would be pointless to use both RTS/CTS and DTR/DSR for flow control signals at the same time. Because of this, DTR and DSR are rarely used for flow control.

**CD** stands for **Carrier Detect**. Carrier Detect is used by a modem to signal that it has a made a connection with another modem, or has detected a carrier tone.

The last remaining line is **RI** or **Ring Indicator**. A modem toggles the state of this line when an incoming call rings your phone.

The Carrier Detect (CD) and the Ring Indicator (RI) lines are only available in connections to a modem. Because most modems transmit status information to a PC when either a carrier signal is detected (i.e. when a connection is made to another modem) or when the line is ringing, these two lines are rarely used.
[Back to Top](http://www.taltech.com/resources/intro-sc.html#top)

--

|  |
| --- |
| **9 to 25 Pin Adapters** |

The following table shows the connections inside a standard 9 pin to 25 pin adapter.

|  |  |
| --- | --- |
| **9-Pin Connector** | **25 Pin Connector** |
| Pin 1 DCD | Pin 8 DCD |
| Pin 2 RD | Pin 3 RD |
| Pin 3 TD | Pin 2 TD |
| Pin 4 DTR | Pin 20 DTR |
| Pin 5 GND | Pin 7 GND |
| Pin 6 DSR | Pin 6 DSR |
| Pin 7 RTS | Pin 4 RTS |
| Pin 8 CTS | Pin 5 CTS |
| Pin 9 RI | Pin 22 RI |

[Back to Top](http://www.taltech.com/resources/intro-sc.html#top)

--

|  |
| --- |
| **Baud vs. Bits per Second** |

The baud unit is named after Jean Maurice Emile Baudot, who was an officer in the French Telegraph Service. He is credited with devising the first uniform-length 5-bit code for characters of the alphabet in the late 19th century. What baud really refers to is modulation rate or the number of times per second that a line changes state. This is not always the same as bits per second (BPS). If you connect two serial devices together using direct cables then baud and BPS are in fact the same. Thus, if you are running at 19200 BPS, then the line is also changing states 19200 times per second. But when considering modems, this isn't the case.

Because modems transfer signals over a telephone line, the baud rate is actually limited to a maximum of 2400 baud. This is a physical restriction of the lines provided by the phone company. The increased data throughput achieved with 9600 or higher baud modems is accomplished by using sophisticated phase modulation, and data compression techniques.
[Back to Top](http://www.taltech.com/resources/intro-sc.html#top)

--

|  |
| --- |
| **Cables, Null Modems, and Gender Changers** |

In a perfect world, all serial ports on every computer would be DTE devices with 25-pin male "D" connectors. All other devices to would be DCE devices with 25-pin female connectors. This would allow you to use a cable in which each pin on one end of the cable is connected to the same pin on the other end. Unfortunately, we don't live in a perfect world. Serial ports use both 9 and 25 pins, many devices can be configured as either DTE or DCE, and - as in the case of many data collection devices - may use completely non standard or proprietary pin-outs. Because of this lack of standardization, special cables called null modem cables, gender changers and custom made cables are often required.
[Back to Top](http://www.taltech.com/resources/intro-sc.html#top)

--

|  |
| --- |
| **Cables Lengths** |

The RS-232C standard imposes a cable length limit of 50 feet. You can usually ignore this "standard", since a cable can be as long as 10000 feet at baud rates up to 19200 if you use a high quality, well shielded cable. The external environment has a large effect on lengths for unshielded cables. In electrically noisy environments, even very short cables can pick up stray signals. The following chart offers some reasonable guidelines for 24 gauge wire under typical conditions. You can greatly extend the cable length by using additional devices like optical isolators and signal boosters. Optical isolators use LEDs and Photo Diodes to isolate each line in a serial cable including the signal ground. Any electrical noise affects all lines in the optically isolated cable equally - including the signal ground line. This causes the voltages on the signal lines relative to the signal ground line to reflect the true voltage of the signal and thus canceling out the effect of any noise signals.

|  |  |  |
| --- | --- | --- |
| **Baud Rate** | **Shielded Cable Length** | **Unshielded Cable Length** |
| 110 | 5000 | 1000 |
| 300 | 4000 | 1000 |
| 1200 | 3000 | 500 |
| 2400 | 2000 | 500 |
| 4800 | 500 | 250 |
| 9600 | 250 | 100 |

--

|  |
| --- |
| **Gender Changers** |

A problem you may encounter is having two connectors of the same gender that must be connected.  You can purchase gender changers at any computer or office supply store for under $5.

***Note:*** The parallel port on a PC uses a 25 pin female connector which sometimes causes confusion because it looks just like a serial port except that it has the wrong gender. Both 9 and 25 pin serial ports on a PC will always have a male connector.
[Back to Top](http://www.taltech.com/resources/intro-sc.html#top)

--

|  |
| --- |
| **Null Modem Cables and Null Modem Adaptors** |

If you connect two DTE devices (or two DCE devices) using a straight RS232 cable, then the transmit line on each device will be connected to the transmit line on the other device and the receive lines will likewise be connected to each other. A Null Modem cable or Null Modem adapter simply crosses the receive and transmit lines so that transmit on one end is connected to receive on the other end and vice versa. In addition to transmit and receive, DTR & DSR, as well as RTS & CTS are also crossed in a Null modem connection.

Null modem adapter are available at most computer and office supply stores for under $5.
[Back to Top](http://www.taltech.com/resources/intro-sc.html#top)

--

|  |
| --- |
| **Synchronous and Asynchronous Communications** |

There are two basic types of serial communications, synchronous and asynchronous. With Synchronous communications, the two devices initially synchronize themselves to each other, and then continually send characters to stay in sync. Even when data is not really being sent, a constant flow of bits allows each device to know where the other is at any given time. That is, each character that is sent is either actual data or an idle character. Synchronous communications allows faster data transfer rates than asynchronous methods, because additional bits to mark the beginning and end of each data byte are not required. The serial ports on IBM-style PCs are asynchronous devices and therefore only support asynchronous serial communications.

Asynchronous means "no synchronization", and thus does not require sending and receiving idle characters. However, the beginning and end of each byte of data must be identified by start and stop bits. The start bit indicate when the data byte is about to begin and the stop bit signals when it ends. The requirement to send these additional two bits cause asynchronous communications to be slightly slower than synchronous however it has the advantage that the processor does not have to deal with the additional idle characters.

An asynchronous line that is idle is identified with a value of 1, (also called a mark state). By using this value to indicate that no data is currently being sent, the devices are able to distinguish between an idle state and a disconnected line. When a character is about to be transmitted, a start bit is sent. A start bit has a value of 0, (also called a space state). Thus, when the line switches from a value of 1 to a value of 0, the receiver is alerted that a data character is about to come down the line.

USB (Universal Serial Bus) is a way of setting up communication between a computer and peripheral devices. USB is intended to replace many varieties of [serial](http://en.wikipedia.org/wiki/Serial_port) and [parallel ports](http://en.wikipedia.org/wiki/Parallel_port). USB can connect [computer peripherals](http://en.wikipedia.org/wiki/Computer_peripheral) such as [mice](http://en.wikipedia.org/wiki/Mouse_%28computing%29), [keyboards](http://en.wikipedia.org/wiki/Computer_keyboard), [PDAs](http://en.wikipedia.org/wiki/Personal_digital_assistant), [gamepads](http://en.wikipedia.org/wiki/Gamepad) and [joysticks](http://en.wikipedia.org/wiki/Joystick), [scanners](http://en.wikipedia.org/wiki/Image_scanner), [digital cameras](http://en.wikipedia.org/wiki/Digital_camera), [printers](http://en.wikipedia.org/wiki/Computer_printer), personal [media players](http://en.wikipedia.org/wiki/Media_player), [flash drives](http://en.wikipedia.org/wiki/USB_flash_drive), and [external hard drives](http://en.wikipedia.org/wiki/External_hard_drive). For many of those devices, USB has become the standard connection method. USB was designed for [personal computers](http://en.wikipedia.org/wiki/Personal_computer), but it has become commonplace on other devices such as PDAs and [video game consoles](http://en.wikipedia.org/wiki/Video_game_console), and as a [power cord](http://en.wikipedia.org/wiki/Power_cord) between a device and an [AC adapter](http://en.wikipedia.org/wiki/AC_adapter) plugged into a [wall plug](http://en.wikipedia.org/wiki/AC_power_plugs_and_sockets) for charging. As of 2008[[update]](http://en.wikipedia.org/w/index.php?title=Universal_Serial_Bus&action=edit), there are about 2 billion USB devices sold per year, and about 6 billion total sold to date.[[1]](http://en.wikipedia.org/wiki/Universal_Serial_Bus#cite_note-0)

The USB 1.0 specification was introduced in 1996. It was intended to replace the multitude of connectors at the back of PCs, as well as to simplify software configuration of communication devices. The original USB 1.0 specification had a data transfer rate of 12 [Mbit](http://en.wikipedia.org/wiki/Mbit)/s.

**USB Overview**

Universal Serial Bus (USB) is a hardware interface for low-speed peripherals such as keyboards,  joysticks, scanners, printers, telephony devices, and MPEG-1 and MPEG-2 digital video. The electrical design of USB  limits the maximum cable length to 5 metres for full-speed devices and up to 3 metres for low-speed devices. Up to 5 hubs can be connected in series to allow for an extended cable run. Up to 127 devices (theoretical limit) can be attached to a single host computer. The hot-swap capability allows a USB device to be plugged in or unplugged without turning the system off. USB devices may be plugged into a USB receptacle on the PC, into a multi-port USB hub or into a USB device that also functions as a hub for other devices.

**Data Transfer**

The Full-speed USB 1.1 maximum bandwidth of 12 Mbits/sec is equivalent to a data transfer speed of 1.5 Mbytes/sec with Hi-speed USB 2.0 dramatically increasing this to 480 Mbits/sec. A Low-speed rate of 1.5Mbits/sec is used for HI (Human Input Devices) such as keyboards, joysticks etc. lControl, Interrupt, Bulk and Isochronous transfer modes are supported.  A USB device indicates its speed by pulling either the D+ or D- line high to 3.3 volts. The pull-up resistors at the device end are also used by the host or hub to detect the presence of a device connected to the port.

Data signals are transmitted on a twisted pair labelled D+ and D− collectively using half-duplex differential signaling to combat the effects of electromagnetic noise on longer lines. D+ and D− usually operate together; they are not separate simplex connections. Transmitted signal levels are 0.0-0.3 volts for low and 2.8-3.6 volts for high. A NRZI (Non Return to Zero Invert) encoding scheme used to send data with a sync field to synchronize the host and receiver clocks

**USB Power**

The USB bus supplies 5V DC regulated power (maximum 500mA)through each port on pins 1 and 4 (pins 1 & 5 on mini-USB). These pins are longer than the data pins to ensure that the power connections mate first and un-mate last. Low-power devices that might normally require a separate AC adapter can therefore be powered via the USB cable, eliminating the need for associated AC adaptors. Bus-powered hubs derive all their power from the USB bus. Powered-hubs are powered from their own AC adapter and provide better power distribution to downstream devices. Port-switching USB hubs isolate all ports from each other so that a faulty device will not cause all other the devices on the same bus to also fail. The specification provides for no more than 5.25 V and no less than 4.35 V between bus positive and negative.